BANNARI AMMAN INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade
Sathyamangalam - 638401 Erode District, Tamil Nadu, India

Name	**SUNIL G**
Roll number	**7376221CS324**
Seat number	**378**
Project number	**31**
Problem statement	**Minutes of Meeting (MoM) Automation**

PROBLEM STATEMENT:

The current manual process of documenting minutes of meetings within our campus community is inefficient and error-prone, causing time-consuming documentation, inconsistent formats, poor accessibility to past records, and delayed dissemination of meeting outcomes. To address these issues, we propose a web-based application using Java to automate the workflow of generating and managing minutes of meetings. This system will streamline documentation, ensure consistency and accuracy, improve accessibility, and expedite the review and approval process, enhancing communication and decision-making across the campus.

Key Objectives:

- **Automate Documentation**: Streamline the process of recording, transcribing, and formatting minutes of meetings to ensure consistency and accuracy..
- **Improve Efficiency**: Reduce the time and effort required to document, review, approve, and distribute minutes, enabling timely dissemination of meeting outcomes.
- **Enhance Accessibility:** Provide a centralized, searchable repository for storing and retrieving minutes of meetings, ensuring easy access to past records.
- **Standardize Formats:** Implement standardized templates for different types of meetings to ensure uniformity in documentation.

- **Facilitate Feedback and Approval:** Implement automated workflows for reviewing, providing feedback, and approving minutes, expediting the finalization process.
- **Integrate Technology**: Utilize Java and related technologies to build a robust, scalable, and secure web-based application tailored to the needs of the campus community.

STACK:

Component	JAVA Tech Stack
Frontend	React js
Backend	Java Spring Boot
Database	MySQL
Authentication	Spring Security with JWT
Styling	CSS/Material-UI/Bootstrap

PROGRESS - TIMELINE:

Phase	Deadline	Status	Notes
Stage 1	03/04/2024	In Progress	Planning and Requirement Gathering
Stage 2		Not Started	Design and Prototyping
Stage 3		Not Started	DB Designing
Stage 4		Not Started	Backend Implementation
Stage 5		Not Started	Testing & Implementation
Stage 6		Not Started	Deployment

PROJECT OVERVIEW: Minutes of Meeting Automation

Purpose

The purpose of this project is to develop an automated Minutes of Meeting (MoM) generation system tailored to the needs of the campus community. The system aims to streamline the process of documenting, reviewing, approving, and disseminating minutes of meetings, ensuring accuracy, consistency, and timely availability of meeting records.

Scope

The project encompasses the design and implementation of a web application using the Java technology stack. It will facilitate meeting scheduling, real-time minute recording, post-meeting documentation, and automated workflows for review and approval. The application will support various meeting types including advisory board, board of study, standing committee, academic counseling, and management meetings, with role-based access control for participants, coordinators, and administrators.

Business Context

The MoM automation system will enhance operational efficiency within the college by automating and streamlining the documentation of meeting minutes. This will reduce administrative overhead, ensure compliance with institutional standards, and provide accurate and accessible meeting records. The system will ultimately contribute to improved communication, decision-making, and accountability across the institution.

Dependencies

Key dependencies for the project include Java for backend development, Spring Boot for building the application, Spring Security for authentication and authorization, Thymeleaf for the frontend, and MySQL for the database. Additional libraries and tools such as Hibernate for ORM, JWT for token-based authentication, and Maven for project management will also be utilized.

Database Management System (DBMS)

The project will use MySQL as the Database Management System. MySQL is a relational database that offers robust data management capabilities, making it ideal for handling structured data related to meeting schedules, participants, minutes, and approvals. It will store user information, meeting details, and approval statuses, ensuring efficient data retrieval and management.

User Personas

- **Participants**: Attend meetings and need access to view scheduled meetings, meeting details, and approved minutes. They require a user-friendly interface to access meeting information and records.
- **Coordinators:** Responsible for scheduling meetings, recording minutes in real-time, and managing the review and approval process. They need tools to efficiently document and distribute meeting minutes.
- **Administrators**: Oversee the entire process, manage user roles and permissions, and ensure compliance with institutional standards. They need a dashboard to monitor system usage, view summaries, and manage user access.

FUNCTIONAL REQUIREMENTS:

User Authentication

Description: Securely authenticate users to ensure that only authorized individuals can access the system.
Features: Login with email and password, role-based access control for participants, coordinators, and administrators, JWT-based authentication for secure sessions.

Meeting Scheduling

Description: Allow coordinators to schedule meetings with necessary details.

Features:

- Forms for meeting type selection (advisory board, board of study, standing committee, academic counseling, management), date and time input, venue selection, participant invitations.
- Real-time validation to ensure no scheduling conflicts and adherence to institutional policies.

Real-Time Minute Recording

Description: Enable coordinators to record meeting minutes in real-time.

Features:

- Tools for documenting discussions, decisions, and action items.
- Predefined templates to ensure consistency and completeness of the recorded minutes.

Post-Meeting Documentation

Description: Facilitate the post-meeting process of documenting and finalizing minutes.

Features:

- Automated transcription and formatting of recorded minutes.
- Options to upload supporting documents and attach them to meeting records.

Review and Approval

Description: Enable designated reviewers to manage and approve meeting minutes.

Features:

- Dashboard for viewing draft minutes, options to approve or request revisions.
- Add comments or remarks for feedback.
- Automated status updates upon approval or request for changes.

User Interface

Description: Provide a user-friendly interface for all user roles to interact with the system.

Features:

- Responsive design for desktops and mobiles.
- Intuitive navigation, clear and organized forms for meeting scheduling and minute recording.
- Accessible design to cater to all users.

Notification System

Description: Keep users informed about the status of their meetings and minutes.

Features:

- Email notifications for meeting invitations, draft submissions, and status changes.
- In-app notifications for pending approvals or updates.
- Customizable alert settings for different user roles.

Reporting

Description: Generate and view reports on meeting minutes and approvals for analysis and record-keeping.

Features:

- Summary reports of meetings held, approval statistics.
- Detailed records by meeting type or department.
- Export options for data analysis and archival purposes.

Minutes of Meeting Generation

Description: Automate the generation of minutes of meetings for different types of meetings.

Features:

- Automated generation of MoM for Board of Studies (BoS), Academic Counseling Meetings (ACM), and Standing Committee Meetings (SCM).
- Templates tailored to each meeting type, ensuring consistent and comprehensive documentation.

PROGRESS AND TOOLS USED:

Stage 1: Planning and Requirement Gathering

Tech Stack: N/A **Tools**:

- **Trello or Jira**: For task management and tracking project progress.
- **Google Docs**: For documentation and collaborative writing.
- **Slack or Microsoft Teams**: For team communication, meetings, and real-time collaboration.

Stage 2: Design and Prototyping

Tech Stack: N/A **Tools**:

- **Figma or Adobe XD**: For creating high-fidelity wireframes and interactive prototypes.
- **Balsamiq**: For creating low-fidelity mockups to visualize the layout and functionality early in the design process.

Stage 3: Database Designing

Tech Stack:

- **MySQL** (Note: The previous mention of MongoDB has been updated to MySQL to align with the Java tech stack).

Tools:

- **MySQL Workbench**: For designing and managing the database schema.
- **Draw.io or Lucidchart**: For visualizing database schema designs and relationships between tables.

Stage 4: Backend Implementation

Tech Stack:

- **Java**
- **Spring Boot**
- **MySQL**

Tools:

- **Postman**: For API testing and debugging.
- **IntelliJ IDEA or Eclipse**: For development in Java.
- **Git**: For version control and collaboration.

Stage 5: Frontend Implementation

Tech Stack:

- **Reactjs**(integrated with Spring Boot)
- **Bootstrap or Material-UI** (for styling and responsive design)

Tools:

- **IntelliJ IDEA or Eclipse**: For development.
- **Figma**: For design integration to ensure the frontend matches the design prototypes.
- **Git**: For version control and collaboration.
- **Axios**: For handling HTTP requests.

Stage 6: Testing & Implementation

Tech Stack:

- **JUnit**: For unit testing in Java.
- **Selenium**: For end-to-end testing.

Tools:

- **Jenkins**: For continuous integration and continuous deployment (CI/CD).
- **Sentry**: For error monitoring and tracking in production.
- **Chrome DevTools**: For debugging frontend issues.

Stage 7: Deployment

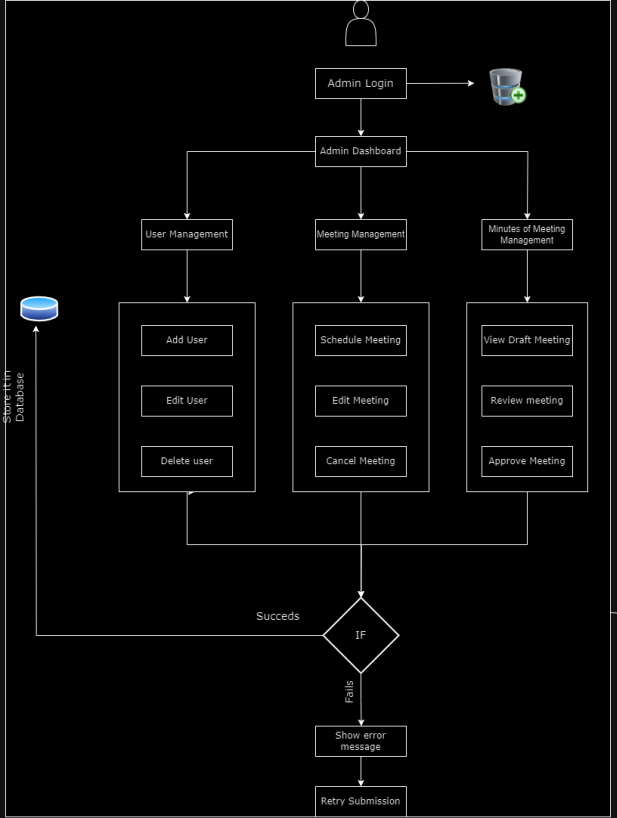
Tech Stack:

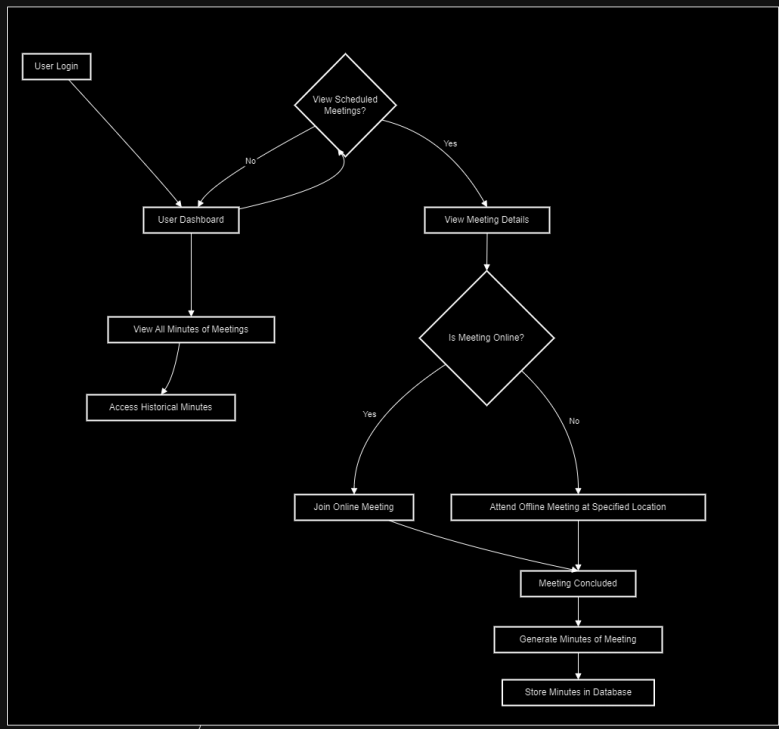
- **AWS**, **Heroku**, or **Vercel**: For hosting and deployment of the web application.

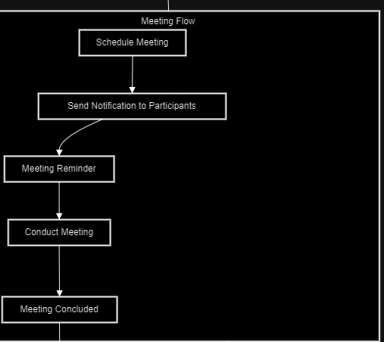
Tools:

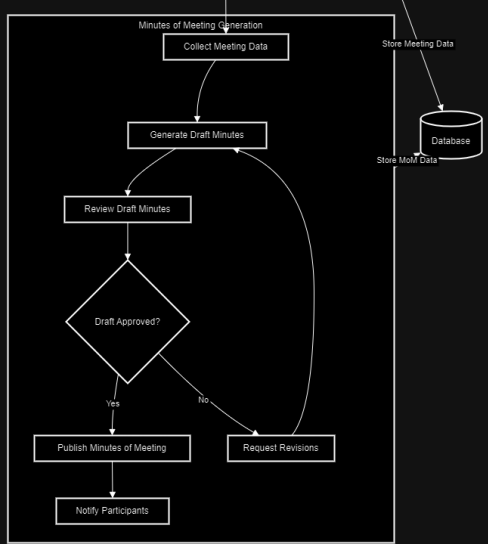
- **GitHub Actions**: For CI/CD pipelines to automate the deployment process.
- **Docker**: For containerization to ensure consistent environments across development, testing, and production.

FlowChart :


Admin Login
Admin Dashboard
User Management
Meeting Management
Minutes of Meeting Management
Add User
Edit User
Delete user
Schedule Meeting
Edit Meeting
Cancel Meeting
View Draft Meeting
Review meeting
Approve Meeting
Store it in Database
Succeds
IF
Fails
Show error message
Retry Submission


User Login
View Scheduled Meetings?
No
Yes
User Dashboard
View Meeting Details
View All Minutes of Meetings
Access Historical Minutes
Is Meeting Online?
Yes
No
Join Online Meeting
Attend Offline Meeting at Specified Location
Meeting Concluded
Generate Minutes of Meeting
Store Minutes in Database


Meeting Flow
Schedule Meeting
Send Notification to Participants
Meeting Reminder
Conduct Meeting
Meeting Concluded


Minutes of Meeting Generation
Collect Meeting Data
Store Meeting Data
Generate Draft Minutes
Database
Store MoM Data
Review Draft Minutes
Draft Approved?
Yes
No
Publish Minutes of Meeting
Request Revisions
Notify Participants

Database